A Learnable Discrete-Prior Fusion Autoencoder with Contrastive Learning for Tabular Data Synthesis

Rongchao Zhang[1], Yiwei Lou[1], Dexuan Xu[2], Yongzhi Cao[1], Hanpin Wang[1], Yu Huang[1,3*]

[1]Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, School of Computer Science, Peking University, Beijing, China
[2]School of Software & Microelectronics, Peking University, Beijing, China
[3]National Engineering Research Center for Software Engineering, Peking University, Beijing, China
{rcz,xudexuan}@stu.pku.edu.cn, cyfqylyw@gmail.com, {caoyz,whpxhy,hy}@pku.edu.cn

Abstract

The actual collection of tabular data for sharing involves confidentiality and privacy constraints, leaving the potential risks of machine learning for interventional data analysis unsafely averted. Synthetic data has emerged recently as a privacy-protecting solution to address this challenge. However, existing approaches regard discrete and continuous modal features as separate entities, thus falling short in properly capturing their inherent correlations. In this paper, we propose a novel contrastive learning guided Gaussian Transformer autoencoder, termed GTCoder, to synthesize photo-realistic multimodal tabular data for scientific research. Our approach introduces a transformer-based fusion module that seamlessly integrates multimodal features, permitting for mining more informative latent representations. The attention within the fusion module directs the integrated output features to focus on critical components that facilitate the task of generating latent embeddings. Moreover, we formulate a contrastive learning strategy to implicitly constrain the embeddings from discrete features in the latent feature space by encouraging the similar discrete feature distributions closer while pushing the dissimilar further away, in order to better enhance the representation of the latent embedding. Experimental results indicate that GTCoder is effective to generate photo-realistic synthetic data, with interactive interpretation of latent embedding, and performs favorably against some baselines on most real-world and simulated datasets.

Introduction

Machine learning has demonstrated immense potential across various industries, and the demand for more sophisticated, cutting-edge AI technology is growing in numerous applications (Patel et al. 2023). Currently, machine learning heavily relies on feature extraction algorithms to extract valuable insights from large-scale datasets. However, the real-world datasets collected in most fields are often disappointing (Price and Cohen 2019), given that these datasets frequently involve confidentiality and privacy concerns. Data isolation and privacy pose significant challenges for AI applications dealing with large-scale data. To solve such challenges, federated learning has provided a well-developed solution, but it is complicated to manage and still faces the risk of data leakage (Bietti et al. 2022). Meanwhile, several approaches (Geng and Viswanath 2016; Wang and Hegde 2019) add random noise to the original data for information exchange, causing the information from the original data to be corrupted. Differential privacy protects personal privacy by adding randomness. Nevertheless, owing to the ever-present irreconcilable conflict between data leakage risk and availability, such approaches still have a loss of raw information, which results in poorly trained AI models (Cheng et al. 2022). The adoption of synthetic data for machine learning has gained significant traction in recent years (Cortés et al. 2022). Researchers have been exploring a well-secured approach to generate synthetic data that closely resembles real data, often referred to as “almost-but-not-quite replica data” (Ganev, Oprisanu, and De Cristofaro 2022). This approach ensures that the synthetic data captures essential characteristics and patterns present in the original dataset while also preserving data privacy and confidentiality. As a result of these efforts, many synthesis data methods have emerged, such as Bayesian network-based (Zhang et al. 2017; Baak et al. 2022), GAN-based (Xu et al. 2019; Esmaeilpour et al. 2022; Xiao, Wu, and Lin 2021), and VAE-based (Xu et al. 2019; Dilokthanakul et al. 2017).

However, tabular data commonly contains both discrete and continuous modal features (Chen 2021), which is complicated to model, thus it poses a great challenge in designing the fusion scheme for model architecture. Existing approaches (Park et al. 2018; Xu et al. 2019; Esmaeilpour et al. 2022) regard discrete and continuous features as separate entities, which have not taken the full capture of their correlations. Recently, transformer based on the self-attention mechanism has shown superiority (Kim et al. 2021) on computer vision and natural language processing tasks. The merits of self-attention mechanism bring a new perspective to the development of feature fusion (Sun et al. 2021). Nevertheless, up until the date of this work, the attempt of extending transformers to capture interactive shared information representations of tabular data remains scarce. Besides, since different discrete features are dependent on each other, how the interaction of different discrete modalities should take place is the key question to answer.

In this paper, we propose a novel framework named GTCoder for synthetic data generation, which plays the

[*] Corresponding author.

strengths of attention mechanism and contrastive learning in latent layer feature generation, achieving better feature embedding. Unlike the transformer (Gorishniy et al. 2021) in classification tasks that learns discriminative information potentially more attentive to subtle differences relevant to categories, we tweak the transformer for the encoder to fuse tabular features and generate latent embeddings, enabling the model to capture the overall data structure as well as its variability. Also, our approach can provide contributions of each feature to the fused feature due to the advantage of the self-attention mechanism. Besides, we combine the lexical information before fusion with the semantic information after fusion, thereby avoiding the loss of raw information. Variational autoencoder (Kingma and Welling 2013) learns a uni-modal Gaussian prior, which is however inadequate for modeling complicated distributions. Therefore, we optimize GMVAE (Cao, Luo, and Klabjan 2021) as framework to learn multi-modal priors from discrete features by using Gaussian mixture encoder, which can generate learnable discrete feature embeddings. Inspired by the work in (Bai, Kong, and Gomes 2022), we introduce contrastive learning to constrain the latent embeddings from discrete features, so that the representations of discrete features are enhanced. GTCoder performs favorably against several baselines on both real-world and simulated datasets, and provides interactive interpretation of latent embedding. Using various ablation experiments, we validate the components of our approach. Our contributions are summarized:

- We propose GTCoder, the transformer-based and contrastive learning-enhanced architecture for tabular data synthesis, thereby obtaining a better privacy protection for scientific research.
- We propose to exploit transformer attention mechanism for tabular feature semantic fusion. It fuses unimodal and multimodal features to generate latent representations in the encoder.
- We introduce a contrastive learning strategy to encourage the similar discrete feature distributions closer while pushing the dissimilar further away, which has a dynamic constraint on representativeness for latent embeddings.

Related Work

Tabular Data Synthesis

Synthetic data has demonstrated robust results in overcoming data limitations for various tasks such as dataset balancing (Xiao, Wu, and Lin 2021), data analysis (Cortés et al. 2022; Zhang et al. 2019; Wang et al. 2019; Lou et al. 2022) and privacy preservation (Faisal et al. 2022; Liu et al. 2022). Nowadays, the main popular methods to synthesize tabular data by using deep learning are based on GANs. A sample balancing technique based on WGAN is proposed in (Xiao, Wu, and Lin 2021) to oversample using a few classes of samples from real biological data. CTGAN is proposed in (Xu et al. 2019) for tabular data synthesis, which has been shown to be better than Bayesian networks. In (Esmaeilpour et al. 2022), a bi-discriminator GAN for synthesizing tabular datasets containing continuous, binary, and discrete columns is presented. Meanwhile, VAE has demonstrated superiority in generative tasks. Previous research (Xu et al. 2019) primarily concentrated on modeling within unimodal potential spaces, overlooking the acquisition of more intricate representations. Recent research (Cao, Luo, and Klabjan 2021) has employed multimodal priors for extension and applied them to open-set recognition. While we optimize GMVAE (Cao, Luo, and Klabjan 2021) as our base framework, it can learn more complex latent representations for synthetic data.

Transformer for Feature Fusion

The transformer was first proposed in (Vaswani et al. 2017) for machine translation, and since then, it has been widely applied to various tasks in the field of natural language processing and extended for computer vision tasks (Qin et al. 2022; Xu et al. 2023). A popular direction explores adopting transformer to multimodal feature fusion. To extract relevance with respect to cross-modal information, a multimodal fusion transformer has been proposed in (Shvetsova et al. 2022), which can process input of any combination of modalities and any length. The work (Zhang et al. 2022a) fuses the multimodal information from the images, audio and text based on transformer, and realize effective affective analysis from different views. In (Bandara and Patel 2022), a multi-scale fusion feature network is proposed by using transformer at different scales of the backbone network, which can capture cross-feature space dependencies. Recently, the work (Bai, Kong, and Gomes 2022) has introduced a simple adaptation of the transformer architecture for tabular data, which outperformed other deep learning methods on most tasks and become a new powerful solution for the field of tabular data classification. However, current tabular transformers are not directly applicable to model data generation. This motivates us to explore the performance of the tabular transformer in autoencoders.

Contrastive Learning

Contrastive learning seeks to learn mutual information by maximizing the similarity between two instances from one class and minimizing the similarity from different classes. Contrastive learning has demonstrated its effectiveness in reinforced feature representation (Carlsson et al. 2021; Pan et al. 2021; Lee and Shin 2022; Zhang et al. 2022b). Semantically similar feature embeddings should have comparable representations. For example, the work (Jian, Gao, and Vosoughi 2022) contrasts examples from text and examples from another modality simultaneously while learning sentence embeddings. In (Yan et al. 2022), a contrastive learning method is exploited in the latent metric space to explore the useful negative correlation hidden in noisy data, which can improve the robustness of DNNs. Similarly, the work (Wang et al. 2021) extends contrastive learning to the multilabel classification task. It selects anchor samples from embeddings of labels and features, clustering related label embeddings together while pushing away from irrelevant embeddings. Different from (Wang et al. 2021), we would like to build a contrastive pipeline for effective latent embedding representations, while leveraging relevance of discrete features from each other.

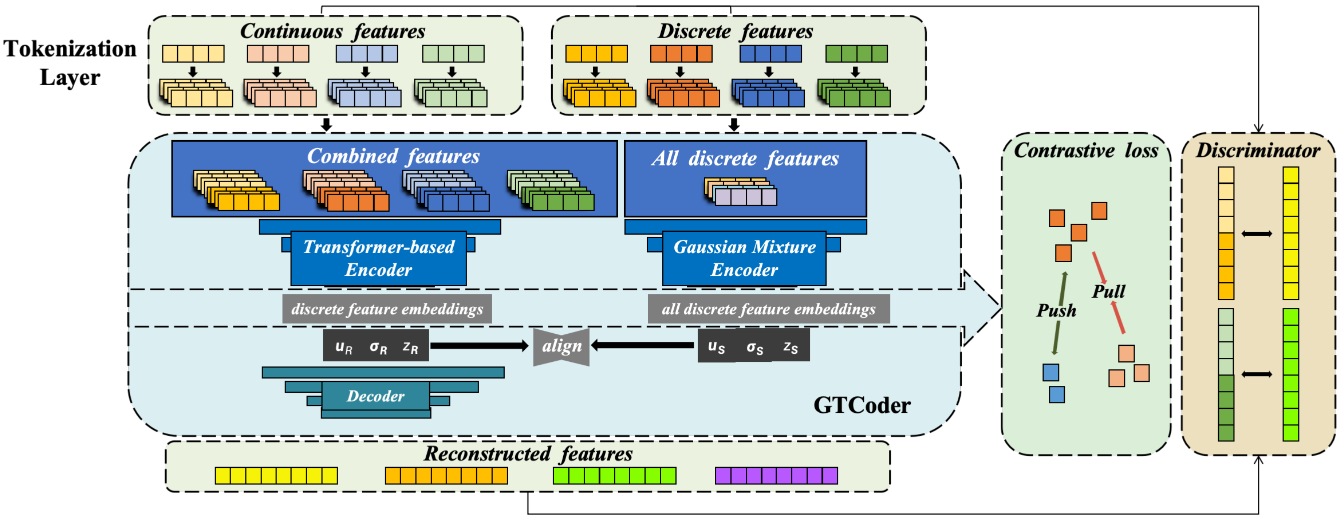


Figure 1: The overall framework of GTCoder. GTCoder follows the main encoder-decoder architecture of GMVAE with a novel Gaussian mixture encoder layer. The transformer-based encoder fuses all features, from which discrete feature embeddings are extracted and contrasted with the embeddings output from the Gaussian mixture encoder.

Methods

The overall architecture of our proposed GTCoder is presented in Fig. 1. Continuous and discrete modalities are fed into a tokenization layer, where the raw inputs are projected to embeddings, followed by GTCoder. There are three components. Transformer-based encoder is used to fuse continuous and discrete features to produce latent embeddings. Gaussian mixture encoder transforms discrete features to learnable embeddings and maps them to a latent space, aligning with the latent space produced by transformer-based encoder. Furthermore, the latent feature embeddings are decoded into reconstruction features by the decoder.

Contrastive learning is optimized to constrain discrete feature embeddings, encouraging similar discrete feature distributions closer while pushing dissimilar further away. This can be seen as an extension of the contrastive mechanism proposed in (Bai, Kong, and Gomes 2022). We describe details about the proposed approach in the following.

Transformer-based Multimodal Fusion Encoder

Let $D = \{(x^{(i)})\}_{i=1}^{N}$ denotes a dataset, where $x^{(i)} = (x_{con}^{(i)}, x_{dis}^{(i)})$ represents continuous features $x_{con}^{(i)} = \{(x_{con,j}^{(i)})\}_{j=1}^{J}$ and discrete features $x_{dis}^{(i)} = \{(x_{dis,t}^{(i)})\}_{t=1}^{T}$ of an object and N denotes the number of objects. Our final goal is to learn an encoder $f_{enc}(x)$ and a decoder $f_{dec}(x)$ to synthesize reconstructed features from D, and the output of $f_{dec}(x)$ is expected to be close to the input D as much as possible. To this end, we introduce a transformer-based fusion architecture as the backbone of the encoder shown in Fig. 2. The encoder transforms discrete and continuous modal features into semantic embeddings respectively and a sentence embedding containing fused information.

We first define a modality-specific tokenization layer that takes as input the raw features and returns a sequence of embeddings to be fed to the transformer. Suppose we have the i-th input data $x^{(i)}$ that contains both continuous and discrete modalities, with the embedding computed as:

$$T_{con,j}^{(i)} = g_{con,j}(x_{con,j}^{(i)}) \in \mathbb{R}^{C}, \tag{1}$$

$$T_{dis,j}^{(i)} = g_{dis,j}(x_{dis,j}^{(i)}) \in \mathbb{R}^{C}, \tag{2}$$

where C denotes the number of tokens, g_{con} and g_{dis} denote the tokenization of continuous and discrete modal features respectively. We apply a linear mapper with reversible mapping of each feature to a C-dimensional vector. This is equivalent to word embedding, which has been widely used in natural language processing (Jiang et al. 2022).

To ensure that output tokens can learn different modal information, [CLS] token is used in the multimodal encoder to extract features for both continuous and discrete tokens. The sequence of input tokens to the transformer follows the below formulation:

$$T_{in}^{(i)} = \left[[\text{CLS}]^{(i)}, T_{con}^{(i)}, T_{dis}^{(i)}\right] \in \mathbb{R}^{(K+1)\times C}, \tag{3}$$

where K is the total number of continuous and discrete features of an object.

For simplicity, we tweak the PreNorm variant (Zhai and Meng 2021), which has been used in tabular data classification (Gorishniy et al. 2021). Specifically, the multi-head-attention module is performed with matrix multiplication of queries, keys, and values in a multi-head manner, which can capture non-local correlation from different modalities:

$$c_i = \text{Attention}(QW_i^{Q}, KW_i^{K}, VW_i^{V}), \tag{4}$$

$$Multi(Q, K, V) = \text{Concat}(c_1, c_2, \cdots, c_h)W^{o}, \tag{5}$$

where Q, K, V are query, key and value matrices respectively. $W_i^{Q} \in \mathbb{R}^{d\times\frac{d}{h}}$, $W_i^{K} \in \mathbb{R}^{d\times\frac{d}{h}}$, $W_i^{V} \in \mathbb{R}^{d\times\frac{d}{h}}$ denote

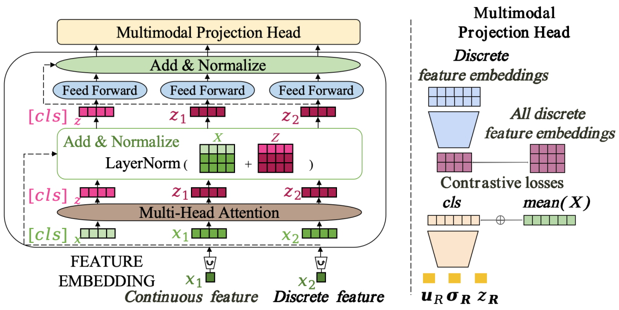


Figure 2: Illustration of the presented transformer-based fusion encoder module.

the transformation matrices of the i-th head and d is the feature dimension of query, key and value. Attention is the self-attention computation function, thus c_i is the output of the i-th head. h is the number of heads and W^o denotes the output transformation matrix. This provides each feature token with strong and direct awareness supervision, making each feature token be able to capture feature-specific information.

The accumulation of embeddings computed by transformers from different layers can learn diverse semantic information (Li et al. 2020). Similarly, we can compute the latent embeddings by summing the [CLS] containing the semantic representation of all features with X containing lexical information, which maximally learns the correlation between multimodal features:

$$\mu = \text{Linear}_{\mu}([\text{CLS}]_{output} + \text{Mean}(x_{con}, x_{dis})), \tag{6}$$

$$\sigma = \text{Linear}_{\sigma}([\text{CLS}]_{output} + \text{Mean}(x_{con}, x_{dis})), \tag{7}$$

where $[\text{CLS}]_{output}$ is the [CLS] token of output layer for transformer and Linear is a fully connected layer.

Contrastive Learning with Learnable Prior

As suggested in (Bai, Kong, and Gomes 2022), using category features to form a Gaussian mixture prior promotes a latent representation meaningful to reconstruct the samples. Consequently, we directly feed discrete features represented as one-hot vectors $x^{(i)}_{dis,j} \in \{0,1\}^L$ to the Gaussian mixture encoder and form several individual latent Gaussian distributions. We activate the positive Gaussian $(x^{(i)}_{dis,jk} = 1)$ and form a Gaussian mixture subspace. The probability density function in the subspace is:

$$p_{\psi}(z|x^{(i)}_{dis}) = \frac{1}{\sum_j \sum_k x^{(i)}_{dis,jk}} \sum_{j=1}^{M} \sum_{k=1}^{L} \mathbb{1}\{x^{(i)}_{dis,jk} = 1\} \mathcal{N}(z|\mu_{jk}, diag(\sigma^2_{jk})) , \tag{8}$$

where $\mathbb{1}$ is the indicator function and μ_{jk}, σ_{jk} are output from the Gaussian mixture encoder. In addition, we also use the weights $w^{(i)}_{dis,j} \in \mathbb{R}^{L\times d}$ from the first fully connected layer as the embeddings for discrete features. Since the embeddings are dynamically updated while training, the learned prior knowledge is also dynamically updated.

The KL-divergence loss function is developed to align the prior with the posterior, which can be written as:

$$\mathcal{L}_{KL} \approx \log q_{\phi}(z^{(i)}_0|x^{(i)}_{con}) - \log p_{\psi}(z^{(i)}_0|x^{(i)}_{dis}), \tag{9}$$

where $z^{(i)}_0$ denotes the latent representation computed by μ and σ of the i-th sample and transformer-based encoder is parameterized by ϕ. Decoder is used to further transform the latent representations to synthesis samples. This process is formulated as:

$$(x^{(i)}|z^{(i)}_0) \sim \mathcal{B}(\mu(z^{(i)}_0; \phi)). \tag{10}$$

To properly learn semantics for feature embeddings, latent representations require implicitly having mutual information existed between feature embeddings. For example, "minor" commonly appears together with "unmarried", while "adult" commonly appears together with "married".

Following the specificity of multimodal projection head from the output of the transformer-based fusion encoder, we extract the latent embeddings for the discrete features to be used as the anchor samples. All discrete feature embeddings are obtained from the Gaussian mixture encoder. We regard positive and negative discrete feature embeddings as positive and negative samples, respectively. More specifically, we define $P^{(i)}_{dis,j} \equiv \{x^{(i)}_{dis,jk} = 1\}^{L}_{k=1}$, where $j \in \{1, \cdots, M\}$. For a batch of samples B, we apply contrastive loss to capture mutual information:

$$\begin{aligned}\mathcal{L}_{CL} = & \frac{1}{|B| \times M} \sum_{x^{(i)} \in B} \sum_{j \in \{1,\cdots,M\}} \frac{1}{|P^{(i)}_{dis,j}|} \sum_{p \in P^{(i)}_{dis,j}} \\ & - \log \frac{\text{sim}(w^f_d, w^l_p)}{\sum_{t \in \{1,\cdots,L\}} \text{sim}(w^f_d, w^l_t)},\end{aligned} \tag{11}$$

where $\text{sim}(u, v) = \exp(\frac{u \cdot v}{\tau})$ is a similarity function that computes the similarity between two feature vectors, and τ denotes a temperature hyperparameter to adjust the softness of the objectives in distinguishing the positive samples from the negative samples. The latent representations for discrete features and the discrete feature embeddings from the Gaussian mixture encoder are denoted as w^f_d and w^l_t, respectively.

Training Scheme

We use reconstruction loss to make the synthetic data look similar to real samples. Let z be a data from a batch of samples B , which has C continuous features. Assume that a token for t can be acquired, denoted by T. Similarly, a reconstructed synthetic data X is defined. The reconstruction loss takes the following form:

$$\begin{aligned}\mathcal{L}_{RE} = & \frac{1}{|B|} \sum_{T \in B} \text{cross\_entropy}(T^{\alpha}, X^{\alpha}) \\ & + \sum_{j \in C} \frac{(T^0_j - \text{t}(X^0_j))^2}{2 \times std^2} + \log(\theta)\end{aligned} , \tag{12}$$

where $\text{cross\_entropy}(T, X)$ is a cross entropy function. For T^0_j and X^0_j, they denote the label encoded column for the

Datasets	**Pregnancy**		**Adult**		**Abalone**		**Ionosphere**		**Agaricus-lepiota**		**Sonar**		**News**
Metrics	**ma-F1**	**we-F1**	**ma-F1**	**we-F1**	**ma-F1**	**we-F1**	**ma-F1**	**we-F1**	**ma-F1**	**we-F1**	**ma-F1**	**we-F1**	**r^2**
Identity	0.66	0.85	0.76	0.83	0.52	0.52	0.86	0.88	1.00	1.00	0.77	0.77	-0.019
CTGAN	0.57	0.71	0.69	0.79	**0.52**	**0.52**	0.48	0.53	0.85	0.85	0.60	**0.62**	-0.035
Bi-discriminator GAN	0.65	0.78	0.69	0.78	0.47	0.47	0.45	0.48	0.88	0.88	0.44	0.47	-0.487
WGAN	0.53	0.79	0.66	0.76	0.43	0.44	0.38	0.38	0.87	0.87	0.44	0.47	-0.437
TVAE	0.57	0.71	0.69	0.79	0.49	0.49	0.48	0.50	0.88	0.88	0.47	0.51	-0.025
LVAE	0.48	0.78	0.66	0.76	0.49	0.49	0.51	0.55	0.92	0.92	0.59	0.61	-0.023
GMVAE	0.55	0.79	0.67	0.77	0.45	0.44	0.41	0.49	0.92	0.92	0.57	0.61	-0.027
GTCoder	**0.66**	**0.83**	**0.70**	**0.80**	**0.52**	0.48	**0.60**	**0.62**	**0.95**	**0.95**	**0.61**	0.61	**-0.020**

Table 1: Comparative results on real-world datasets. We compare our method with CTGAN (Xu et al. 2019), Bi-discriminator GAN (Esmaeilpour et al. 2022), WGAN (Xiao, Wu, and Lin 2021), TVAE (Xu et al. 2019), LVAE (Sønderby et al. 2016), and GMVAE (Dilokthanakul et al. 2017).

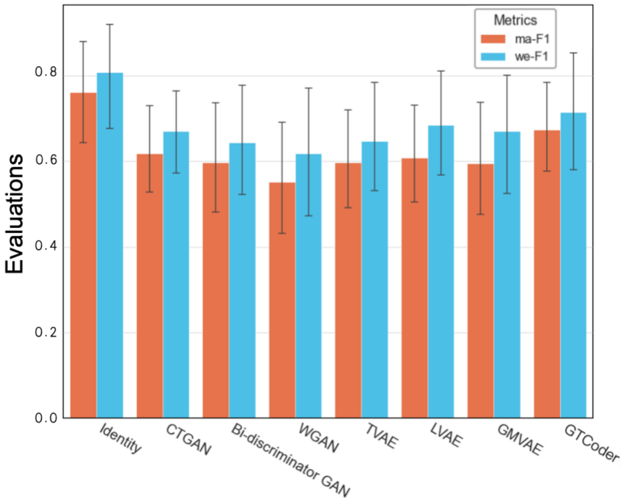


Figure 3: Results on real-world datasets (higher is better).

j-th continuous feature of real data and reconstructed data, respectively. The columns other than the label encoded column are denoted as T^α and X^α. We have θ as a random parameter output with the decoder, and $\mathrm{t}(X)$ is a $\tanh$ activation function. The synthetic data can be generated by inverse decoding of the reconstructed token.

The overall per-sample objective for training the entire GTCoder model end-to-end is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{KL} + \lambda_2 \mathcal{L}_{CL} + \lambda_3 \mathcal{L}_{RE}, \qquad (13)$$

where λ_i balances the three losses.

Interpretability for GTCoder

In attention mechanism, output x^i_{output} of $token^i$ is computed by weighting the value vector v^j of all tokens. The weight α_{ij} of v^j can be got from the attention matrix. Thus, with the weight α_{ij}, the contribution of $token^i$ to $token^j$ can be measured. We choose the row corresponding to [CLS] from the attention matrix and regard it as the degree of contribution of all tokens to the fused representation. By applying the attention mechanism on latent representation generation, which features play important roles in representation learning can be directly observed to guide the synthesis of more effective data.

Experiments

Experimental Setup

Datasets. We use six commonly available tabular datasets from UCI (Dua and Graff 2017) with various numbers of discrete (contained labels) and continuous features, including Adult, News, Abalone, Ionosphere, Agaricus-lepiota and Sonar. Adult contains 9 discrete and 6 continuous features. News contains 46 continuous and 14 discrete features, which makes the dataset more complex. We selected Abalone, which has a smaller and simpler number of features, containing only 2 discrete and 7 continuous features. Ionosphere has a small sample size but features with a certain complexity, which includes 32 continuous and 3 discrete features, therefore it better reflects the performance of the models for extracting features. A characteristic of Sonar is that it contains only 60 continuous features and 1 label, which can make the data synthesis particularly challenging. Conversely, Agaricus-lepiota contains only 23 discrete features, thus it better shows the ability of an algorithm to learn relevance combinations of discrete features.

Combining medical data with machine learning fully exploits the value of medical data. However, both publishing data and training data in machine learning may reveal the privacy of patients, thus more effective privacy protection methods are urgently needed to ensure the security of released medical data (Su et al. 2021). To demonstrate the effectiveness of our approach in privacy protection, we also conduct experiments on a medical clinical dataset (Pregnancy). The Pregnancy dataset includes various physical indicators of the patients and a label for whether they had a

Datasets		Grid		Gridr		Ring	
Metrics		L_{syn}	L_{test}	L_{syn}	L_{test}	L_{syn}	L_{test}
GAN-based methods	CTGAN	-1.68	**-2.88**	-1.75	**-2.88**	-1.61	**-2.89**
	Bi-discriminator GAN	-1.56	**-2.88**	-1.56	-2.89	-1.65	-2.95
	WGAN	**-1.43**	-2.89	**-1.25**	-2.95	**-1.19**	-3.17
VAE-based methods	TVAE	-1.07	-4.36	-0.99	-3.90	**-0.79**	-4.99
	LVAE	-0.71	-4.59	-1.11	-9.03	-0.96	-9.41
	GMVAE	-2.13	-3.27	-1.15	-4.63	-1.84	-2.96
	GTCoder	**-0.68**	**-3.11**	**-0.67**	**-3.25**	-1.51	**-2.89**

Table 2: Comparison results on the grid, gridr, and ring. L_{syn} represents the likelihood fitness on simulations for the synthetic dataset of each method. Re-fitting simulations with synthetic data, and then the likelihood fitness on the test sets to obtain L_{test}.

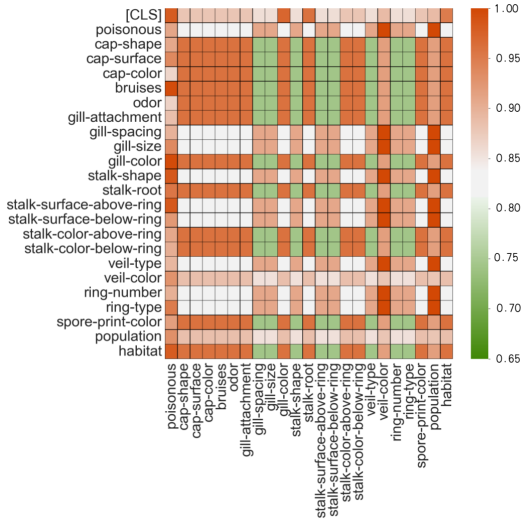


Figure 4: Interpretation of feature-feature interactions. Green and red colors indicate the least and most significant contributions, respectively.

Datasets	CL	Prior	TFT	ma-F1	we-F1	AVG.Δ
Ionosphere	-	-	-	0.479	0.497	-
	-	✓	-	0.410	0.497	-3.45%
	-	-	✓	0.483	0.572	3.95%
	-	✓	✓	0.562	0.621	10.35%
	✓	✓	✓	**0.603**	**0.623**	**12.5%**
Agaricus-lepiota	-	-	-	0.882	0.883	-
	-	✓	-	0.920	0.920	3.75%
	-	-	✓	0.896	0.897	1.40%
	-	✓	✓	0.936	0.936	5.35%
	✓	✓	✓	**0.953**	**0.953**	**14.2%**
Pregnancy	-	-	-	0.573	0.710	-
	-	✓	-	0.547	0.793	2.85%
	-	-	✓	0.596	0.796	5.45%
	-	✓	✓	0.643	0.817	8.85%
	✓	✓	✓	**0.657**	**0.830**	**10.2%**

Table 3: Ablation study for each component of our model.

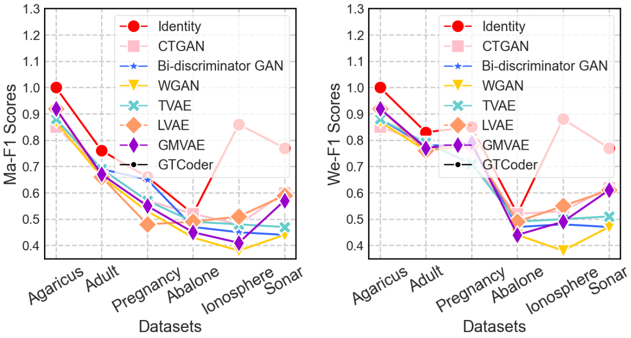
miscarriage, which would not be publicly available because it involves the privacy of the patients.

Moreover, we construct three Gaussian mixture simulations to represent known joint distributions. Three simulated datasets were generated by sampling from them separately, called grid, gridr, and ring.

Comparison Methods. As for synthetic data quality, we compare GTCoder with state-of-the-art deep learning methods, including three GAN-based methods (CTGAN, Bi-discriminator GAN, WGAN) and three VAE-based methods (TVAE, LVAE, GMVAE).

Implementation Details. To fairly compare each method, we train all methods using the same size epoch and batch. All real datasets are separated into testing sets called T_{test} (20%) and training sets called T_{train} (80%). As discrete feature encoder and decoder, we simply followed (Bai, Kong, and Gomes 2022) which composed by fully connected networks. Through conducting empirical analysis, we set $\lambda_1 = 1$, $\lambda_2 = 0.5$, and $\lambda_3 = 0.1$. The temperature parameter τ for contrastive learning in GTCoder is set to 0.95.

Evaluation metrics. For the real datasets, we evaluate each method by comparing the performance of machine learning, which are trained on the synthetic datasets. More specifically, we use T_{train} to train a generative model and generate a synthetic dataset with the same size as T_{train}. Then, we train classifiers or regression models using the synthetic dataset, and evaluate them using T_{test}. In the classification tasks, we select well-performing decision tree, random forest and adaboost and compute the mean macro-F1 (ma-F1) and mean weighted-F1 (we-F1) scores of them as evaluation metrics. And for the regression tasks, we select the r^2 of Ridge, Linear and GradientBoosting Regression for evaluation. For the simulated datasets, we utilize the synthetic dataset to compute likelihood fitness (L_{syn}) on the simulation. Next, we retrain a simulation using the synthetic dataset and use T_{test} to compute likelihood fitness (L_{test}) on the simulation.



(a) Macro-F1 scores.

(b) Weighted-F1 scores.

Figure 5: Performance evaluated for all methods.

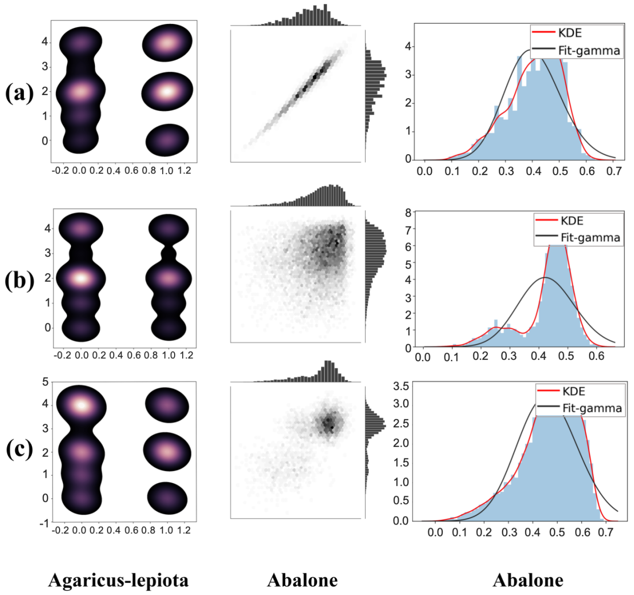


Figure 6: Distribution of synthetic features on the Agaricus-lepiota and Abalone sets. (a) Real data (b) TVAE (c) Ours.

Experimental Results

Evaluation on real-world datasets. In Table 1 and Fig. 3, we evaluate each tabular data synthesis method on all real-world datasets. It is shown that GTCoder outperforms other methods on most datasets. To further compare the performance of each method, we also assess the synthetic data distribution in Fig. 6. Our method excels in generating synthetic data that maintains a high level of consistency with the distribution of real data, ensuring reliable and meaningful analysis. CTGAN is still a fine benchmark method, yet GMVAE and LVAE have already surpassed GAN-based methods on several datasets. Significantly, GTCoder achieves particularly competitive performance on Agaricus-lepiota, while slightly underperforming CTGAN on Abalon and Sonar, which further demonstrates that our approach is more effective for processing datasets with numerous discrete features.

Evaluation on simulated datasets. Further, we report the performance of each generation method on the simulated dataset sampled from the simulation in Table 2. Obviously, GTCoder shows comparatively superior performance, however minority performance is slightly underperformed by CTGAN. The simple feature composition of the simulated dataset and lack of discrete features are possible factors, which present no strengths for our method. Despite these limitations, our method demonstrates considerable potential and effectiveness in various data synthesis tasks, paving the way for further advancements and refinement in the field of synthetic data generation.

Interpretability. Fig. 4 shows an explanation for the interactions of each feature on Agaricus-lepiota. It shows that gill-color, poisonous and habitat have contributed most to the fused features, which explains that several of them are more contributing to synthetic data generation. Poisonous is the label for mushroom species classification, and information of external morphological features can be provided by gill-color. As well, habitat is also an extremely significant feature, as the environment a mushroom grows in can have an influence on its growth condition and adaptability to the external environment. Such features have significant contributions to discriminate whether mushrooms are poisonous or not, thus they have higher weights to fused latent features. The attention weights from transformer can help us to understand the essence of datasets clearly, which can provide us with more precise and reliable synthesis results.

Ablation Studies

We conduct ablation studies to evaluate the effectiveness of the three modules for our approach. The ablation studies demonstrate that our GTCoder consistently outperforms other models by a large margin, with an average improvement in assessment metrics of more than 10%. Although the existence of learnable prior is present, our method performs slightly lower in the absence of contrastive learning, indicating that learnable prior and contrastive learning are remarkably crucial components for GTCoder, which also shows the integration model has superiority.

Conclusion

In this paper, we propose a new approach named GTCoder for multimodal tabular data synthesis, which realizes data privacy protection in scientific research. Our approach contains a multimodal fusion encoder and a learnable prior discrete feature encoder, which can fuse features efficiently and learn rich knowledge. Moreover, we introduce a contrastive learning strategy to constrain latent embedding representations for discrete features. Extensive experiments on most real-world and simulated datasets demonstrate superior performance for the proposed approach.

Acknowledgments

This paper was supported by National Key Research and Development Program of China (2021YFF1201100), National Natural Science Foundation of China under Grants (62172016 and 61932001) and Beijing Nova Program.

References

Baak, M.; Brugman, S.; Fridman Rojas, I.; Dalmeida, L.; E.Q. Urlus, R.; and Oger, J.-B. 2022. Synthsonic: Fast, Probabilistic modeling and Synthesis of Tabular Data. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 4747–4763. PMLR.

Bai, J.; Kong, S.; and Gomes, C. P. 2022. Gaussian Mixture Variational Autoencoder with Contrastive Learning for Multi-Label Classification. In *Proceedings of the 39th International Conference on Machine Learning*, 1383–1398. PMLR.

Bandara, W. G. C.; and Patel, V. M. 2022. HyperTransformer: A Textural and Spectral Feature Fusion Transformer for Pansharpening. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1757–1767.

Bietti, A.; Wei, C.-Y.; Dudik, M.; Langford, J.; and Wu, S. 2022. Personalization Improves Privacy-Accuracy Tradeoffs in Federated Learning. In *Proceedings of the 39th International Conference on Machine Learning*, 1945–1962. PMLR.

Cao, A.; Luo, Y.; and Klabjan, D. 2021. Open-Set Recognition with Gaussian Mixture Variational Autoencoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8): 6877–6884.

Carlsson, F.; Gyllensten, A. C.; Gogoulou, E.; Hellqvist, E. Y.; and Sahlgren, M. 2021. Semantic Re-tuning with Contrastive Tension. In *International Conference on Learning Representations*.

Chen, Y. 2021. *Research on deep probability generation model based on variational inference*. Publishing House of Electronics Industry.

Cheng, A.; Wang, J.; Zhang, X. S.; Chen, Q.; Wang, P.; and Cheng, J. 2022. DPNAS: Neural Architecture Search for Deep Learning with Differential Privacy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6): 6358–6366.

Cortés, A.; Rodríguez, C.; Vélez, G.; Barandiarán, J.; and Nieto, M. 2022. Analysis of Classifier Training on Synthetic Data for Cross-Domain Datasets. *IEEE Transactions on Intelligent Transportation Systems*, 23(1): 190–199.

Dilokthanakul, N.; Mediano, P. A. M.; Garnelo, M.; Lee, M. C.; Salimbeni, H.; Arulkumaran, K.; and Shanahan, M. 2017. Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. *arXiv preprint arXiv:1611.02648*.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. https://archive-beta.ics.uci.edu.

Esmaeilpour, M.; Chaalia, N.; Abusitta, A.; Devailly, F.-X.; Maazoun, W.; and Cardinal, P. 2022. Bi-Discriminator GAN for Tabular Data Synthesis. *Pattern Recogn. Lett.*, 159(C): 204–210.

Faisal, F.; Mohammed, N.; Leung, C. K.; and Wang, Y. 2022. Generating Privacy Preserving Synthetic Medical Data. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–10.

Ganev, G.; Oprisanu, B.; and De Cristofaro, E. 2022. Robin Hood and Matthew Effects: Differential Privacy Has Disparate Impact on Synthetic Data. In *Proceedings of the 39th International Conference on Machine Learning*, 6944–6959. PMLR.

Geng, Q.; and Viswanath, P. 2016. The Optimal Noise-Adding Mechanism in Differential Privacy. *IEEE Transactions on Information Theory*, 62(2): 925–951.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting Deep Learning Models for Tabular Data. In *Advances in Neural Information Processing Systems*, 18932–18943.

Jian, Y.; Gao, C.; and Vosoughi, S. 2022. Non-Linguistic Supervision for Contrastive Learning of Sentence Embeddings. In *Advances in Neural Information Processing Systems*, 35533–35548.

Jiang, T.; Jiao, J.; Huang, S.; Zhang, Z.; Wang, D.; Zhuang, F.; Wei, F.; Huang, H.; Deng, D.; and Zhang, Q. 2022. PromptBERT: Improving BERT Sentence Embeddings with Prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 8826–8837.

Kim, K.; Wu, B.; Dai, X.; Zhang, P.; Yan, Z.; Vajda, P.; and Kim, S. 2021. Rethinking the Self-Attention in Vision Transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 3065–3069.

Kingma, D. P.; and Welling, M. 2013. Auto-Encoding Variational Bayes. *CoRR*, abs/1312.6114.

Lee, K.; and Shin, J. 2022. RényiCL: Contrastive Representation Learning with Skew Rényi Divergence. In *Advances in Neural Information Processing Systems*, 6463–6477.

Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; and Li, L. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9119–9130.

Liu, F.; Cheng, Z.; Chen, H.; Wei, Y.; Nie, L.; and Kankanhalli, M. 2022. Privacy-Preserving Synthetic Data Generation for Recommendation Systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, 1379–1389. ISBN 9781450387323.

Lou, Y.; Huang, Y.; Xing, X.; Cao, Y.; and Wang, H. 2022. MTS-LSTDM: multi-time-scale long short-term double memory for power load forecasting. *Journal of Systems Architecture*, 125: 102443.

Pan, T.; Song, Y.; Yang, T.; Jiang, W.; and Liu, W. 2021. VideoMoCo: Contrastive Video Representation Learning With Temporally Adversarial Examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11205–11214.

Park, N.; Mohammadi, M.; Gorde, K.; Jajodia, S.; Park, H.; and Kim, Y. 2018. Data Synthesis Based on Generative Adversarial Networks. *Proc. VLDB Endow.*, 11(10): 1071–1083.

Patel, D.; Lin, S.; Shah, D.; Jayaraman, S.; Ploennigs, J.; Bhamidipati, A.; and Kalagnanam, J. 2023. AI Model Factory: Scaling AI for Industry 4.0 Applications. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(13): 16467–16469.

Price, W.; and Cohen, I. 2019. Privacy in the age of medical big data. *Nature Medicine*, 25(1): 37–43.

Qin, Y.; Lou, Y.; Huang, Y.; Chen, R.; and Yue, W. 2022. An Ensemble Deep Learning Approach Combining Phenotypic Data and fMRI for ADHD Diagnosis. *Journal of Signal Processing Systems*, 94(11): 1269–1281.

Shvetsova, N.; Chen, B.; Rouditchenko, A.; Thomas, S.; Kingsbury, B.; Feris, R.; Harwath, D.; Glass, J.; and Kuehne, H. 2022. Everything at Once – Multi-modal Fusion Transformer for Video Retrieval. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19988–19997.

Sønderby, C. K.; Raiko, T.; Maaløe, L.; Sønderby, S. K.; and Winther, O. 2016. Ladder Variational Autoencoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 3745–3753.

Su, J.; Cao, Y.; Chen, Y.; Liu, Y.; and Song, J. 2021. Privacy Protection of Medical Data in Social Network. *BMC Medical Informatics and Decision Making*, 21(Suppl 1): 286.

Sun, P.; Zhang, W.; Wang, H.; Li, S.; and Li, X. 2021. Deep RGB-D Saliency Detection with Depth-Sensitive Attention and Automatic Multi-Modal Fusion. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1407–1417.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Wang, B.; and Hegde, N. 2019. *Privacy-Preserving Q-Learning with Functional Noise in Continuous Spaces*. Red Hook, NY, USA: Curran Associates Inc.

Wang, P.; Han, K.; Wei, X.-S.; Zhang, L.; and Wang, L. 2021. Contrastive Learning based Hybrid Networks for Long-Tailed Image Classification. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 943–952.

Wang, Q.; Gao, J.; Lin, W.; and Yuan, Y. 2019. Learning From Synthetic Data for Crowd Counting in the Wild. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8190–8199.

Xiao, Y.; Wu, J.; and Lin, Z. 2021. Cancer Diagnosis Using Generative Adversarial Networks Based on Deep Learning from Imbalanced Data. *Comput. Biol. Med.*, 135(C).

Xu, D.; Zhu, H.; Huang, Y.; Jin, Z.; Ding, W.; Li, H.; and Ran, M. 2023. Vision-knowledge fusion model for multi-domain medical report generation. *Information Fusion*, 97: 101817.

Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; and Veeramachaneni, K. 2019. *Modeling Tabular Data Using Conditional GAN*. Red Hook, NY, USA: Curran Associates Inc.

Yan, J.; Luo, L.; Xu, C.; Deng, C.; and Huang, H. 2022. Noise Is Also Useful: Negative Correlation-Steered Latent Contrastive Learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 31–40.

Zhai, Z.; and Meng, X. 2021. *Deep Learning: Theory, Methods, and Pytorch Practice*. Beijing: Tsinghua University Press.

Zhang, J.; Cormode, G.; Procopiuc, C. M.; Srivastava, D.; and Xiao, X. 2017. PrivBayes: Private Data Release via Bayesian Networks. *ACM Trans. Database Syst.*, 42(4).

Zhang, L.; Gonzalez-Garcia, A.; van de Weijer, J.; Danelljan, M.; and Khan, F. S. 2019. Synthetic Data Generation for End-to-End Thermal Infrared Tracking. *IEEE Transactions on Image Processing*, 28(4): 1837–1850.

Zhang, W.; Qiu, F.; Wang, S.; Zeng, H.; Zhang, Z.; An, R.; Ma, B.; and Ding, Y. 2022a. Transformer-based Multimodal Information Fusion for Facial Expression Analysis. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2427–2436.

Zhang, Y.; Zhang, R.; Mensah, S.; Liu, X.; and Mao, Y. 2022b. Unsupervised Sentence Representation via Contrastive Learning with Mixing Negatives. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 11730–11738.